

## Program szkolenia:

# Klasyczne i współczesne style architektoniczne

## Informacje:

<b>Nazwa:</b>	<b>Klasyczne i współczesne style architektoniczne</b>
<b>Kod:</b>	<b>arch-patterns-Style</b>
<b>Kategoria:</b>	Wzorce architektoniczne
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

Szkolenie ma na celu zapoznanie uczestników z metodami tworzenia, zarządzania i dokumentowania architektury systemów w oparciu o ideę stylu architektonicznych.

Styl architektoniczny w połączeniu z taktykami, wzoracmi i heurystykami tworzy język, który ułatwia komunikowanie i opisywanie tak złożonego zagadnienia, jakim jest architektura systemu.

Podczas szkolenia uczestnicy zapoznają się z wybranymi stylami architektonicznymi. Poznają ich zastosowania, wady i zalety. Nauczą się budować własne style służące do opisu architektury ich systemów.

**Program szkolenia jest ogólną ramą - konkretne style wybieramy podczas analizy przed-szkoleniowej.**

## Zalety szkolenia:

- Zawiera wzorce i najlepsze praktyki
- Aspekty architektury aplikacji i systemu

## Szczegółowy program:

### 1. Co mamy na myśli mówiąc "Architektura"

1.1. Architektura jako składowa procesu realizacji wymagań

1.2. Architektura systemu

1.2.1. Element

1.2.1.1. przetwarzanie

1.2.1.2. dane

1.2.1.3. łączenie

1.2.2. Forma dla elementu

1.2.2.1. priorytety

1.2.2.2. relacje

1.2.2.3. wagi

1.2.3. Uzasadnienie

1.2.3.1. motywacja

1.2.3.2. weryfikacja

### 2. Jak odróżnić dobrą architekturę od złej

2.1. Realizacja wymagań nie-funkcjonalnych

2.2. Realizacja wymagań funkcjonalnych

2.3. Metryki

2.3.1. Wydajność

2.3.2. Rozszerzalność

2.3.3. Testowalność

2.3.4. Bezpieczeństwo

2.3.5. Koszt utrzymania

### 3. Jak sprawnie dokumentować architekturę

#### 3.1. Widoki architektoniczne

##### 3.1.1. Kontekst i grupa docelowa

##### 3.1.2. Metoda C4

##### 3.1.3. Tworzenie języka wizualnego

### 4. Style architektoniczne

#### 4.1. Wstęp

##### 4.1.1. Co to jest styl architektoniczny

##### 4.1.2. Styl jak zbiór formalnej aranżacji elementów

##### 4.1.3. Styl jako forma reużycia i enkapsulacji wiedzy

#### 4.2. Dla systemów scentralizowanych/monolitycznych

##### 4.2.1. Pipes and Filters

##### 4.2.2. Hierarchiczne

###### 4.2.2.1. Ogólna charakterystyka

###### 4.2.2.2. Warstwowe

###### 4.2.2.3. Porty i Adaptery

##### 4.2.3. Dziedzinowe

###### 4.2.3.1. DDD

###### 4.2.3.2. DSL

##### 4.2.4. Persystencji

###### 4.2.4.1. Repository

###### 4.2.4.2. Collection

###### 4.2.4.3. Event Sourcing

###### 4.2.4.4. Active Record

#### 4.3. Dla systemów rozproszonych

#### 4.3.1. SOA/Microservice

4.3.1.1. Orkiestracja vs Choreografia

4.3.1.2. Wyznaczanie granicy serwisu

4.3.1.3. Spójność danych

4.3.1.4. Zastosowania

#### 4.3.2. Message Driven

4.3.2.1. Event vs Command

4.3.2.2. Broker

4.3.2.3. Bus

4.3.2.4. Listener

4.3.2.5. Saga

#### 4.3.3. REST

4.3.3.1. Poziomy dojrzałości

4.3.3.2. REST jako zbiór Ograniczeń

4.3.3.3. Zastosowania

### 5. Jak stworzyć własny styl architektoniczny

5.1. Taktyki

5.2. Heurystyki

5.3. Wzorce

5.4. Zasady

### 6. Jak utrzymać architekturę w ryzach

6.1. Problemy

6.1.1. Kruchość

6.1.2. Erozja

6.1.3. Dryft

## 6.2. Sposoby pielęgnacji architektury

### 6.2.1. Retrospektywy architektoniczne

### 6.2.2. Architektura w code review

### 6.2.3. Informative workspace

### 6.2.4. Architektura i SCRUM